LoHoSearch: Benchmarking Long-Horizon Search Agents Beyond the Human Difficulty Ceiling

Anonymous ACL submission

Abstract

Search agent benchmarks exemplified by BrowseComp have rapidly saturated over the past year, with the strongest models surpassing 90% accuracy. Since these benchmarks are predominantly human-authored, annotators lack a global perspective on entity statistics and cannot systematically maximize search space size and structural complexity. This creates a difficulty ceiling that is hard to break. To address this, we introduce **LoHoSearch** (**Lo**ng-**Ho**rizon **Search** Agents), a challenging benchmark comprising 544 human-verified questions across 11 domains. LoHoSearch is constructed via an automated pipeline built upon a knowledge graph covering over 7 million Wikipedia entities, which selects relations with large search spaces and assembles them into structurally complex questions with KG-verified unique answers. Our evaluation demonstrates that even the strongest model achieves only 34.74% accuracy, and existing context management strategies (best +6.8%) yield far smaller gains than on prior benchmarks. LoHoSearch provides a more demanding standard for evaluating long-horizon reasoning and context management in search agents.

1 Introduction

Since April 2025, challenging yet easily verifiable benchmarks, exemplified by BrowseComp (Wei et al., 2025), have rapidly become the de facto standard for measuring search agent capabilities. Yet, as Figure 1 illustrates, model performance on BrowseComp has soared from 30% to over 90% in barely ten months, and the benchmark is quickly losing its discriminating power (Anthropic, 2026a). The root cause is that these benchmarks are predominantly human-authored: annotators tend to choose entities and relations they are familiar with, which typically have high popularity and direct connections, causing most questions to be answerable within only a few retrieval steps. This forms a difficulty ceiling that is hard to raise further, and as model capabilities continue to advance, this trend will only intensify.

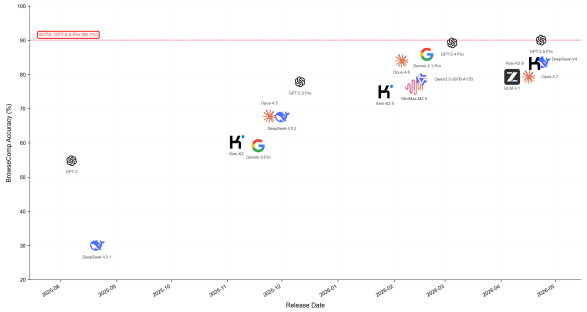

Figure 1: BrowseComp accuracy progression from August 2025 to May 2026 across major model families.

The difficulty of search problems is determined by two core factors: (1) the search space size per constraint, i.e., the number of candidate entities satisfying a single condition. A larger search space forces the agent to verify and eliminate more candidates. (2) The structural complexity, i.e., the number of constraints that must be jointly satisfied to uniquely identify the answer. Higher structural complexity means more constraints must be checked to rule out each candidate, substantially raising the overall solving difficulty. While reasoning depth (the number of knowledge hops required to reach the final answer) also contributes to difficulty, it is the easiest to control and has been well-addressed by existing benchmarks (Trivedi et al., 2022; Krishna et al., 2025). When both search space size and structural complexity are large, the agentic search process becomes substantially longer, placing higher demands on both reasoning and context management. However, human annotators lack a global perspective on entity statistics and cannot systematically maximize difficulty along both dimensions.

To address this, we introduce LoHoSearch,

a benchmark constructed through an automated pipeline grounded in a knowledge graph. Starting from Wikipedia, we build a large-scale knowledge graph spanning over 7 million entities, select relations with genuinely large search spaces under a global view, and assemble them into structurally complex subgraphs whose answers are KG-verified uniqueness. Each subgraph is subsequently verbalized into a natural-language question by a language model and undergoes multiple rounds of automated verification and human review to ensure correctness and answer uniqueness. The resulting benchmark comprises 544 human-verified questions across 11 topical domains. Our main contributions are as follows:

- We propose a knowledge-graph-based automated QA construction pipeline that systematically controls search space size and structural complexity, breaking through the difficulty ceiling of human authoring.
- We introduce the LoHoSearch benchmark. Even the strongest model achieves only 34.7%, with correct trajectories requiring 1.7× more tool calls than BrowseComp, establishing a more discriminative evaluation standard for search agents.
- Our benchmark reveals the limitations of existing context management strategies in high-difficulty scenarios. The best strategy yields only a 6.8% improvement, far below gains on existing benchmarks, demonstrating that LoHoSearch serves as a more challenging testbed for future research.

2 Data Synthesis

Our pipeline proceeds through four stages (illustrated in Figure 2): knowledge graph construction, subgraph sampling, QA generation and verification, and post filtering with human review.

2.1 Knowledge Graph Construction

We construct a knowledge graph from the full English Wikipedia dump: each page corresponds to an entity (node), and hyperlinks within the page body pointing to other Wikipedia pages serve as directed edges. We define each entity's type as its Wikidata (Vrandečić and Krötzsch, 2014) P31 (`instance_of`) class and its popularity as in-degree, both for use in subsequent stages. The resulting knowledge graph contains approximately 7.62 million entities and 265 million directed edges.

2.2 Subgraph Sampling

We employ two complementary subgraph structures: tree-structured and graph-structured. The difficulty of the tree structure mainly arises from the size of the search space, while the graph structure further increases structural complexity through cyclic dependencies and cross-constraints among entities. Both structures require all constituent entities to have low popularity and moderate page lengths, ensuring that entities cannot be easily inferred. Additionally, we balance the type distribution of answer entities during sampling to ensure diversity across topical domains.

We first define the search space of a relation. Given a directed edge from entity A to entity B, its search space is defined as:

$$\mathcal{S}(A \to B) = \{e \mid \text{type}(e) = \text{type}(A), \; (e \to B) \in \mathcal{G}\} \tag{1}$$

That is, in the knowledge graph $\mathcal{G}$, the set of all entities that share the same type as A and also have a directed edge to B. A larger search space means more candidate entities satisfy the given relational constraint, making it harder for the agent to identify A via this relation.

2.2.1 Tree-Structured Subgraph Sampling

The tree structure uses a low-popularity entity as the root node (i.e., the answer), which connects to multiple intermediate entities, each of which further connects to several leaf nodes. Sampling proceeds layer by layer:

First-layer expansion. From the root's relations, we select N edges pointing to intermediate entities, subject to:

- The search space size of each relation $|\mathcal{S}| > \tau$;
- The intersection of candidate sets for any $N-1$ relations has size > 1. That is, the answer is no longer uniquely determined if we remove any single relation, ensuring every relation is necessary;
- The intersection of candidate sets across all N relations equals exactly $\{\text{root}\}$, guaranteeing KG-level uniqueness of the answer.

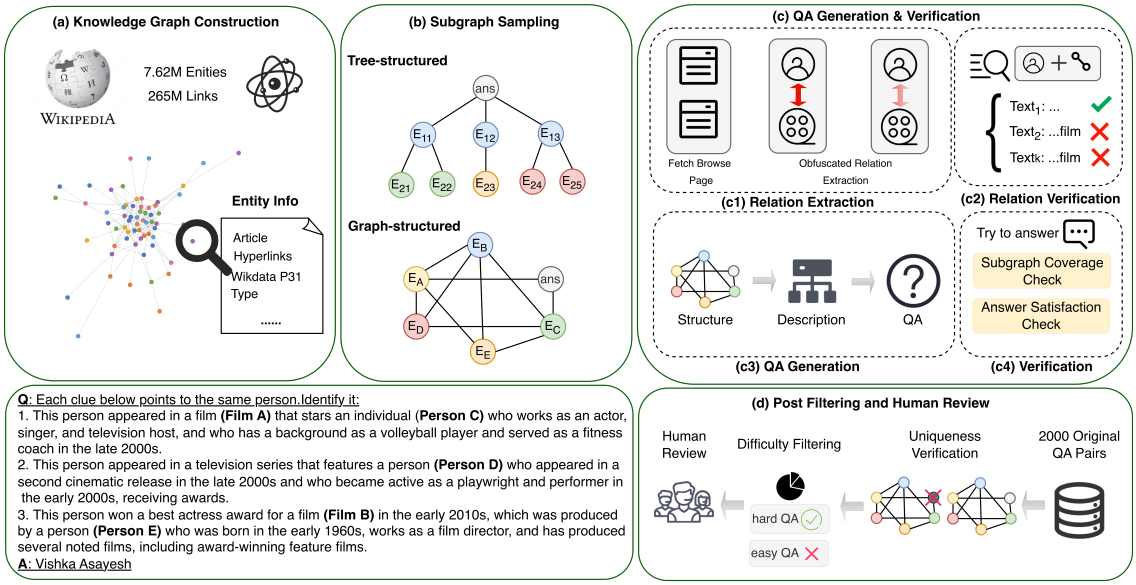

Figure 2: Overview of the LoHoSearch pipeline.

Second-layer expansion. For each intermediate entity, we select 1 to M edges pointing to leaf nodes, subject to:

- The search space size of each relation $|\mathcal{S}| > \tau$;
- The intersection of candidate sets across the M relations has size > 1, ensuring the intermediate entity itself cannot be directly inferred;
- We refer to candidates in this intersection (other than the current intermediate entity) as pseudo-candidates. We further require that no pseudo-candidate, when combined with the remaining intermediate entities, uniquely determines the answer—this preserves answer uniqueness.

In practice, we set $N = 3$, $M = 2$, and $\tau = 3$.

2.2.2 Graph-Structured Subgraph Sampling

Unlike the hierarchical expansion of tree structures, graph-structured subgraphs contain extensive cross-edges among entities and may form cycles, making the problem constraints non-decomposable into independent sub-problems. During sampling, we first select a low-popularity entity as the seed (i.e., the answer), then greedily expand until the subgraph reaches a maximum of 10 entities: at each step, we prioritize the candidate with the most edges to the current subgraph and whose corresponding edges have the largest search space. The subgraph must satisfy node type diversity, sufficient edge count, and connectivity.

After construction, uniqueness is verified via exhaustive backtracking search: we search for another set of entities in the full graph that satisfies the same entity types and directed adjacency relations; if none exists, uniqueness is confirmed. Additionally, we require the seed entity to have a sufficient number of confounding candidates of the same type that are connected to entities of all neighbor types in the subgraph, preventing brute-force solving via type enumeration.

2.3 QA Generation and Verification

This stage converts each sampled subgraph into a natural-language question. Specifically, for each edge in the subgraph, a language model extracts an obfuscated description of the relation from the source entity's Wikipedia page; for leaf nodes in tree structures, 1–2 additional property descriptions are extracted.

After extraction, we apply search-based verification to ensure sufficient obfuscation: each relation must be neither directly locatable via search engines nor inferable by an LLM. To prevent multiple relations from becoming easy to infer when combined, we perform joint verification on all relations of the same entity.

Once verification passes, we assemble all relations and properties into a structured description

with entity names hidden, and have an LLM convert it into a natural-language question. The generated question undergoes two rounds of automated validation:

- Subgraph coverage check: verifies that the question faithfully covers all relations and properties from the input subgraph, with no omissions, additions, or distortions.
- Answer satisfaction check: a search agent verifies that the ground-truth answer indeed satisfies all conditions stated in the question.

All LLM-based steps in this stage are performed using DeepSeek-V3.2 (DeepSeek-AI et al., 2025).

2.4 Post Filtering and Human Review

After the QA generation stage, we subject all questions to multiple rounds of filtering:

Uniqueness verification. Although the subgraph sampling stage enforces structural uniqueness, the transformation from subgraph to natural language may introduce ambiguity. We deploy multiple search agents of different capability levels to independently attempt each question, collect candidate answers, and automatically judge whether any candidate satisfies all conditions. Questions for which an alternative valid answer is found are filtered out.

Difficulty filtering. To calibrate the difficulty of the benchmark, we have a DeepSeek-V3.2-powered search agent attempt each question multiple times independently. Questions that are answered correctly across multiple trials are filtered out, retaining only those that pose genuine search difficulty.

Human review. After all automated filtering, the remaining questions are submitted to professional annotators for manual verification. Annotators evaluate each question along multiple dimensions such as answer correctness, answer uniqueness, logical coherence across conditions, language fluency, and information redundancy, ensuring that the final questions are both rigorous and natural.

2.5 Data Statistics

Table 1 summarizes the key statistics of LoHoSearch. The dataset comprises 544 human-verified questions. Graph-structured subgraphs are notably denser than tree-structured ones, with more nodes and nearly twice as many edges, reflecting their higher structural complexity. As shown in Figure 3, the questions span 11 topical domains, including Music, Geography & Places, Film & Television, and Sports, ensuring broad coverage across knowledge areas.

	Tree	Graph	Total
Questions	282	262	544
Avg. Nodes	7.9	10.0	8.9
Avg. Edges	6.9	12.5	9.6

Table 1: Dataset statistics of LoHoSearch.

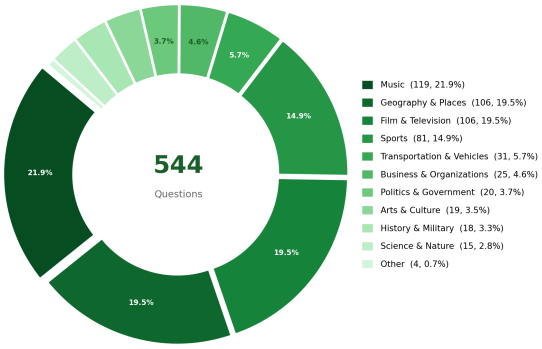

Figure 3: Domain distribution of LoHoSearch. The dataset consists of 544 samples spanning 11 categories.

Regarding quality assurance, 75.5% of all automatically constructed questions passed human review directly, 22.3% were accepted after minor wording adjustments by annotators (e.g., correcting unnatural phrasing or removing redundant qualifiers), and only 2.2% were discarded due to critical issues such as logical inconsistencies, demonstrating the high generation quality of the automated pipeline. In terms of answer uniqueness, 70.8% of questions are confirmed by annotators to have a definitively unique answer. For the remaining 29.2%, annotators were unable to conclusively rule out alternatives, yet found no substitute candidates after thorough search. We note that as question difficulty increases, verifying uniqueness itself becomes extremely challenging even for human annotators, further attesting to the inherent complexity of LoHoSearch questions.

3 Experiments

3.1 Experimental Settings

We select the best-performing model from each major model family based on their results on

Table 2: Performance (%) on LoHoSearch. Score reports the average correct ratio on the 544-sample dataset. Calibration Error measures the confidence calibration of each model. Best in **bold**. †Indicates models that encountered service instability or safety refusals during evaluation.

Model	Reasoning	Source	LoHoSearch Score (%)	Calibration Error (%)
GPT-5.5 (OpenAI, 2026)	N	Closed	**34.74**	48
DeepSeek-V4-Pro† (DeepSeek-AI, 2026)	Y	Open	15.99	35
Claude-Opus-4.6 (Anthropic, 2026b)	N	Closed	15.62	31
Kimi-K2.6† (Moonshot-AI, 2026)	Y	Open	15.53	56
Gemini-3.1-Pro (DeepMind, 2026)	Y	Closed	13.32	72
GLM-5.1 (Zhipu-AI, 2026)	Y	Open	12.77	58
Claude-Opus-4.7 (Anthropic, 2026c)	N	Closed	10.29	35
DeepSeek-V4-Flash (DeepSeek-AI, 2026)	Y	Open	10.02	48
LongCat-Flash-Thinking-2601 (LongCat-Team et al., 2026)	Y	Open	9.74	76
MiniMax-M2.7 (MiniMax, 2026b)	Y	Open	2.48	59
MiniMax-M2.5 (MiniMax, 2026a)	Y	Open	2.29	54

BrowseComp, which are then designated as our evaluation targets; the vast majority are the latest releases within their respective families. The full list of evaluated models is detailed in Table 2.

Each model is equipped with two tools: (1) `search`, which performs keyword queries using traditional search engines such as Google; and (2) `browse`, which retrieves detailed content from one or more specified web pages given their URLs. We adopt the same system prompt as BrowseComp to instruct all models. For models that support both thinking and non-thinking modes or adjustable thinking effort, we adopt their official default settings. To ensure fair comparison across models with potentially different optimal temperature configurations, we uniformly set the temperature to 1.0, which also encourages diverse search strategies. The context window is uniformly set to 200K tokens, with 184K allocated for input and 16K for output.

Evaluation. We adopt an LLM-based automated evaluation approach. Specifically, we compute accuracy once using the BrowseComp grading prompt with GPT-4.1 (OpenAI, 2025) as the judge, and a second time using the SimpleQA (Wei et al., 2024) grading prompt with Qwen2.5-32B (Yang et al., 2025) as the judge. The final score is the average of the two accuracy values. We find that a single grading combination may be either overly strict or overly lenient, and averaging across two complementary combinations best approximates the true performance of each model.

3.2 Main Results

As shown in Table 2, GPT-5.5 achieves the highest LoHoSearch Score of 34.74%, substantially outperforming all other models. DeepSeek-V4-Pro, Claude-Opus-4.6, and Kimi-K2.6 demonstrate comparable performance with scores ranging from 15.53% to 15.99%, while the remaining models all score below 14%. Overall, no model exceeds 35% accuracy, confirming that LoHoSearch poses a substantial challenge to current state-of-the-art search agents. During evaluation, we observed that Kimi-K2.6 and DeepSeek-V4-Pro encountered safety refusals and service instability issues, which may have affected their scores; results impacted by these issues are denoted with † in Table 2.

Regarding confidence calibration, we adopt the calibration error metric following the methodology of BrowseComp-ZH (Zhou et al., 2025). We note that some models did not strictly follow the system-prompt format (e.g., omitting the Confidence field), introducing noise into the calibration results. Despite this, models evaluated on our benchmark exhibit consistently high calibration errors, indicating considerable uncertainty when confronted with these challenging queries. This further corroborates the difficulty of LoHoSearch.

Table 3: Ablation study on various context management strategies based on DeepSeek-V4-Flash. Best in **bold**.

Strategy	BrowseComp	LoHoSearch
Baseline	58.84	10.02
w/ Summary	63.90	11.31
w/ Discard-all	67.10	12.41
w/ Summary + Verify	67.81	15.35
w/ Discard-all + Verify	**72.87**	**16.82**

3.3 Context Management

To investigate how existing context management strategies perform under the extended reasoning

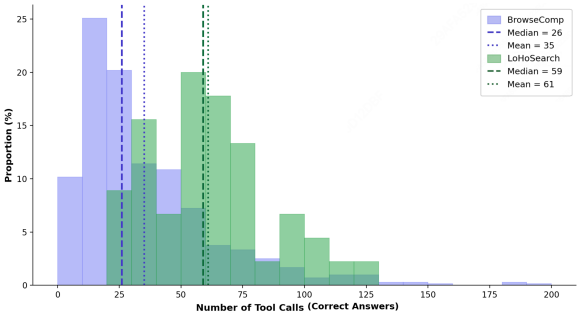

Figure 4: Distribution of the number of tool calls (correct answers) for BrowseComp and LoHoSearch. LoHoSearch requires significantly more tool calls to reach correct answers compared to BrowseComp.

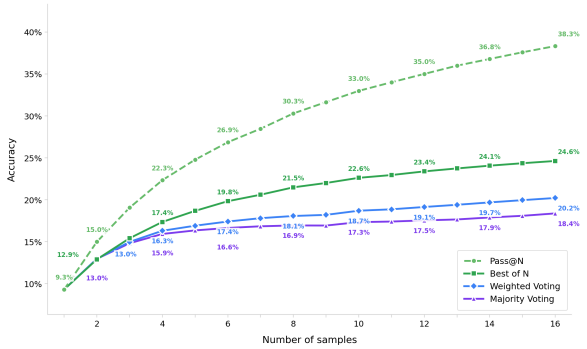

Figure 5: Illustration of pass@N and three answer aggregation strategies (majority voting, weighted voting, and best-of-N) under parallel sampling.

demands of LoHoSearch, we conduct experiments on DeepSeek-V4-Flash with a standard ReAct framework (Yao et al., 2022) as the baseline. We then evaluate two context management strategies triggered when token usage exceeds 80% of the context window: (1) Summary (Wu et al., 2025), which compresses the current trajectory and re-initiates the search. (2) Discard-all (DeepSeek-AI et al., 2025), which discards all previous tool calls and restarts with only the original query. Additionally, we incorporate a Verify module that leverages the query and the most recent reasoning steps to judge whether all query conditions are satisfied before committing to a final answer.

As shown in Table 3, the best-performing combination (Discard-all + Verify) achieves 16.82%, an absolute gain of only 6.8% over the baseline. Notably, the same strategies yield a 14.03% gain on BrowseComp, yet deliver only marginal improvements on LoHoSearch. This discrepancy suggests that the large search space and high structural complexity of our benchmark demand substantially longer reasoning chains, exposing fundamental limitations in current context management approaches. These results also suggest that our benchmark provides a more rigorous testbed for the development of next-generation context management techniques.

3.4 Further Analyses

3.4.1 Difficulty Analysis

We further analyze the difficulty gap between BrowseComp and LoHoSearch using DeepSeek-V4-Flash as the probe model. In terms of accuracy, DeepSeek-V4-Flash achieves 58.84% on BrowseComp but only 10.02% on LoHoSearch (Table 2). Regarding the distribution of tool calls in correct trajectories, as illustrated in Figure 4, solving queries in our benchmark requires substantially more tool calls compared to BrowseComp. Specifically, the mean number of tool calls increases from 35 to 61 (a 74% relative increase), while the median rises from 26 to 59, confirming that LoHoSearch demands substantially more multi-step reasoning and retrieval.

Furthermore, graph-structured questions achieve an accuracy of only 8.01%, compared to 11.89% for tree-structured ones. This gap is primarily attributable to the cyclic dependencies and cross-constraints in graph structures that prevent problem decomposition, indicating that structural complexity, as a difficulty factor independent of search space size, further increases solving difficulty.

3.4.2 Parallel Sampling

To investigate the performance upper bound achievable through repeated sampling on our benchmark, we conduct a parallel sampling analysis following the scoring formulation of BrowseComp. Specifically, we sample 16 independent responses from DeepSeek-V4-Flash and evaluate three answer aggregation strategies: majority voting, weighted voting, and best-of-N selection.

As shown in Figure 5, the pass@N metric improves substantially with increasing sample count, rising from 9.3% at N=1 to 38.3% at N=16, indicating considerable headroom for improvement through repeated sampling. Among the three aggregation strategies, best-of-N, which selects the answer with the highest model confidence score, achieves the strongest performance at 24.6%. This demonstrates that confidence-based selection is more effective than voting-based approaches and yields performance closer to the theoretical

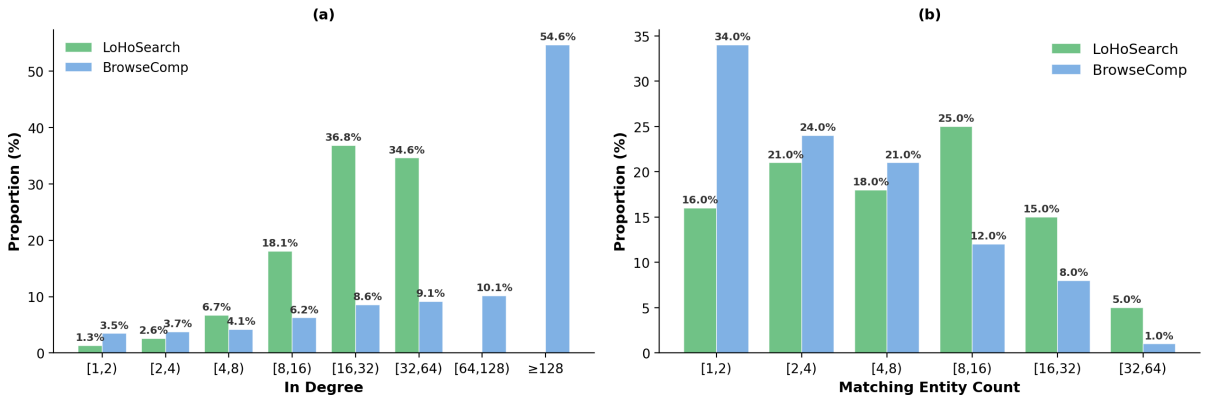

Figure 6: Analysis of hidden entity popularity and search space size. (a) In-degree distribution of hidden entities in BrowseComp and LoHoSearch, where higher in-degree indicates greater entity popularity and easier inferability. BrowseComp contains significantly more high-popularity hidden entities than LoHoSearch. (b) Under the same popularity constraint, LoHoSearch exhibits a substantially larger relational search space, demonstrating the necessity of a knowledge graph for systematically constructing high-difficulty questions.

pass@N upper bound.

3.4.3 Necessity of the Knowledge Graph

We analyze the hidden entities in questions from BrowseComp and LoHoSearch, i.e., entities that the agent must infer on its own during the solving process, to demonstrate the necessity of a knowledge graph for constructing truly challenging questions. Specifically, we focus on the hidden entities directly associated with the answer in each question. For LoHoSearch, hidden entities are obtained directly from the subgraph structure data; since human-authored benchmarks inherently lack such structural annotations, we parse hidden entities from search agents' correct trajectories, and for questions that agents failed to answer correctly, we use DeepSeek-V3.2 to reverse-engineer them from the answer and question text. Both extraction methods share the same objective: identifying intermediate entities that require reasoning during the solving process. After obtaining the hidden entities, we further retrieve their associated information in the knowledge graph, including entity popularity and the search space size of relations.

The results are shown in Figure 6. First, the popularity of hidden entities in BrowseComp is substantially higher than in LoHoSearch (Figure 6a). Although BrowseComp also aims to construct challenging questions, the lack of a global perspective in human annotation makes it difficult to precisely control entity popularity, resulting in hidden entities that tend to be relatively easy to infer. Second, even under the same popularity constraint, the relational search space size of entities in LoHoSearch is significantly larger than in BrowseComp (Figure 6b), meaning that inferring new entities from a given entity is considerably harder. The above analysis shows that human construction has clear limitations in both entity popularity and search space size, and that a knowledge graph is a necessary foundation for systematically constructing high-difficulty questions.

4 Related Work

4.1 Search Agent Benchmarks

Multi-hop question answering benchmarks established the foundation for evaluating complex reasoning over documents. HotpotQA (Yang et al., 2018) contains 113K Wikipedia-based questions covering bridge and comparison reasoning. It was the first dataset to require sentence-level supporting evidence. Building on this, 2WikiMultiHopQA (Ho et al., 2020) extended coverage to four reasoning types across two Wikipedia sources with chain-level evidence annotations. MuSiQue (Trivedi et al., 2022) composed single-hop sub-questions bottom-up into 25K questions of 2–4 hops and used unanswerable variants to resist shortcuts. These benchmarks share two constraints: they operate in closed-domain settings that do not reflect open-web agent use, and they rely on manual construction that limits scale.

Tool-augmented language models shifted evaluation toward the open web. GAIA (Mialon et al., 2024) contains 466 questions at three difficulty levels. FRAMES (Krishna et al., 2025) provides 824 multi-hop questions for RAG pipelines. These

benchmarks bring tool use into scope but still rely on fixed, manually curated question sets. They cannot scale beyond annotation budgets, and they cannot be refreshed once published.

BrowseComp (Wei et al., 2025) serves as the most directly comparable benchmark for our study. It contains 1266 questions designed to be difficult to retrieve but easy to verify, crafted by domain experts through multiple rounds of verification and adversarial filtering. At the time of release, OpenAI Deep Research scored 51.5%, and human testers 33.3%. Despite its high quality, its reliance on manual construction introduces clear limitations: (i) construction costs grow linearly with scale; (ii) answer uniqueness relies on human verification without formal guarantees; (iii) difficulty calibration depends on expert judgment. BrowseComp-ZH (Zhou et al., 2025), which extends the benchmark to Chinese with 289 questions, further illustrates the prohibitive cost of cross-lingual expansion under manual construction.

Recent benchmarks further push difficulty beyond BrowseComp: DeepSearchQA (Gupta et al., 2026) introduces 900 causal-chain tasks requiring exhaustive answer lists, and WideSearch (Wong et al., 2026) evaluates large-scale information collection where even the best system achieves only 5% success rate. However, both rely on manual construction and frame difficulty through multi-subtask composition rather than maximizing the intrinsic difficulty of individual questions. In contrast, LoHoSearch systematically controls search space size and structural complexity at the single-question level.

In summary, existing benchmarks face three key limitations: (i) manual construction that cannot scale; (ii) answer uniqueness without formal guarantees; (iii) difficulty calibration dependent on human judgment. **LoHoSearch** addresses all three through a knowledge-graph-driven automated pipeline.

4.2 QA Generation for Search Agents

Synthetic data methods for search agents focus on training stronger agents rather than evaluating them. WebShaper (Tao et al., 2026) formalizes information seeking into four stages (query formulation, search execution, page browsing, and information synthesis) and repurposes HotpotQA questions to construct agent trajectories. WebSailor (Li et al., 2025) introduces a multi-round progressive search framework and achieves superhuman performance on BrowseComp. WebSailor-V2 (Li et al., 2026) combines synthetic data with scalable reinforcement learning under a progressive difficulty curriculum, significantly narrowing the gap between open-source and proprietary agents.

These works focus on leveraging existing QA datasets as training signals to improve agent performance. In contrast, LoHoSearch takes a fundamentally different approach: We build a large-scale knowledge graph and sample structurally complex subgraphs for QA synthesis. Because the difficulty can be systematically escalated by adjusting search space size and structural complexity parameters, the benchmark keeps pace with the rapid self-improvement cycle of modern agents rather than being left behind by it.

Knowledge-graph-based QA generation constitutes a closely related line of research. KNIGHT (Amanlou et al., 2026) builds topic-specific knowledge graphs as compressed, reusable representations and generates multiple-choice questions at varying difficulty levels through graph traversal. GraphGen (Chen et al., 2025) builds fine-grained knowledge graphs from source text and generates QA pairs at multiple granularities via multi-hop neighborhood sampling.

LoHoSearch shares the knowledge-graph-driven generation philosophy with these works but differs in two key respects. First, LoHoSearch uses the graph primarily to guarantee answer validity through subgraph structural uniqueness. Second, rather than probing static knowledge, LoHoSearch prompts an LLM to extract obfuscated relation descriptions from Wikipedia and applies a search verification step to ensure that no question can be resolved through simple retrieval. Together, these choices decouple the knowledge graph from question content and answer storage, extending KG-driven evaluation from closed-domain knowledge assessment to open-web browsing agent evaluation.

5 Conclusion

We present LoHoSearch, a benchmark constructed via a knowledge-graph-driven pipeline that systematically controls search space size and structural complexity. The resulting questions surpass the difficulty ceiling of human annotation, and experiments confirm that all evaluated models struggle while current context management strategies yield limited gains. LoHoSearch provides a discriminative testbed for advancing search agent capabilities.

Limitations

LoHoSearch currently covers only English. Since our pipeline is language-agnostic by design, we plan to release multilingual variants in future work. Additionally, the released evaluation set is a static snapshot that may become susceptible to contamination or temporal drift over time. On the evaluation side, all models are assessed under the same search-and-browse toolset with a fixed context window, and answer correctness relies on LLM-based judges that may introduce noise on ambiguous edge cases.

Beyond these aspects, two further limitations stem from the construction pipeline itself. First, answer uniqueness is verified within our knowledge graph; for the 29.2% of questions that human annotators could not conclusively confirm as uniquely answerable (§2.4), alternative answers outside the KG cannot be ruled out, though no substitute candidate was found after thorough search. Second, difficulty filtering relies on a single LLM (DeepSeek-V3.2) as the calibration agent, which may introduce family-specific bias in which questions survive filtering. We plan to address these aspects through periodic regeneration, broader tool settings, more diverse calibration agents, and improved judging in subsequent iterations.

Ethical considerations

This research is conducted in full accordance with the EMNLP Code of Ethics. All datasets used in this work were acquired by strictly following their respective licensing agreements and usage guidelines, with no violation of privacy or data protection standards. We have made deliberate efforts throughout the study to examine and mitigate any potential biases or discriminatory effects that may arise in both the data collection and model development processes. No personally identifiable information was included in any part of our experiments, thereby ensuring the protection of individual privacy and security. We are committed to maintaining transparency, reproducibility, and academic integrity across all stages of this research.

References

Mohammad Amanlou, Erfan Shafiee Moghaddam, Mahdi Nouri, Yasaman Amou Jafary, Farhan Farsi, and Behnam Bahrak. 2026. KNIGHT: Knowledge graph-driven multiple-choice question generation with adaptive hardness calibration. In *Proceedings of the Conference on Parsing and Linguistic Theories (CPAL)*.

Anthropic. 2026a. System card: Claude Mythos Preview.

Anthropic. 2026b. System card: Claude Opus 4.6.

Anthropic. 2026c. System card: Claude Opus 4.7.

Zihong Chen, Wanli Jiang, Jinzhe Li, Zhonghang Yuan, Huanjun Kong, Wanli Ouyang, and Nanqing Dong. 2025. GraphGen: Enhancing supervised fine-tuning for llms with knowledge-driven synthetic data generation. *Preprint*, arXiv:2505.20416.

Google DeepMind. 2026. Model card: Gemini 3.1 Pro.

DeepSeek-AI. 2026. DeepSeek-V4: Towards highly efficient million-token context intelligence.

DeepSeek-AI, Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenhao Xu, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, and 245 others. 2025. DeepSeek-V3.2: Pushing the frontier of open large language models. *Preprint*, arXiv:2512.02556.

Nikita Gupta, Riju Chatterjee, Lukas Haas, Connie Tao, Andrew Wang, Chang Liu, Hidekazu Oiwa, Elena Gribovskaya, Jan Ackermann, John Blitzer, Sasha Goldshtein, and Dipanjan Das. 2026. DeepSearchQA: Bridging the comprehensiveness gap for deep research agents. *Preprint*, arXiv:2601.20975.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Satyapriya Krishna, Kalpesh Krishna, Anhad Mohananey, Steven Schwarcz, Adam Stambler, Shyam Upadhyay, and Manaal Faruqui. 2025. Fact, fetch, and reason: A unified evaluation of retrieval-augmented generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4745–4759, Albuquerque, New Mexico. Association for Computational Linguistics.

Kuan Li, Zhongwang Zhang, Huifeng Yin, Rui Ye, Yida Zhao, Liwen Zhang, Litu Ou, Ding-Chu Zhang, Xixi Wu, Xinmiao Yu, Jialong Wu, Xinyu Wang, Zile Qiao, Zhen Zhang, Yong Jiang, Pengjun Xie, Fei Huang, Zhi-Qin John Xu, Shuai Wang, and 2 others. 2026. WebSailor-V2: Bridging the chasm to proprietary agents via synthetic data and scalable reinforcement learning. In *The Fourteenth International Conference on Learning Representations*.

Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, Weizhou Shen, Junkai Zhang, Dingchu Zhang, Xixi Wu, Yong Jiang, Ming Yan, Pengjun Xie, Fei Huang, and Jingren Zhou. 2025. WebSailor: Navigating super-human reasoning for web agent. *Preprint*, arXiv:2507.02592.

LongCat-Team, Anchun Gui, Bei Li, Bingyang Tao, Bole Zhou, Borun Chen, Chao Zhang, Chen Gao, Chen Zhang, Chengcheng Han, and 1 others. 2026. LongCat-Flash-Thinking-2601 technical report. *arXiv preprint arXiv:2601.16725*.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2024. GAIA: a benchmark for general AI assistants. In *The Twelfth International Conference on Learning Representations*.

MiniMax. 2026a. MiniMax M2.5: Built for real-world productivity.

MiniMax. 2026b. MiniMax M2.7: Early echoes of self-evolution.

Moonshot-AI. 2026. Kimi K2.6: Advancing open-source coding.

OpenAI. 2025. Introducing GPT-4.1 in the API.

OpenAI. 2026. Introducing GPT-5.5.

Zhengwei Tao, Jialong Wu, Wenbiao Yin, Pu Wu, Junkai Zhang, Baixuan Li, Haiyang SHEN, Kuan Li, Liwen Zhang, Xinyu Wang, Wentao Zhang, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. 2026. WebShaper: Agentically data synthesizing via information-seeking formalization. In *The Fourteenth International Conference on Learning Representations*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledge base. *Communications of the ACM*, 57:78–85.

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. 2024. Measuring short-form factuality in large language models. *arXiv preprint arXiv:2411.04368*.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. 2025. BrowseComp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*.

Ryan Wong, Jiawei Wang, Junjie Zhao, Li Chen, Yan Gao, Long Zhang, Xuan Zhou, Zuo Wang, Kai Xiang, Ge Zhang, Wenhao Huang, Yang Wang, and Ke Wang. 2026. WideSearch: Benchmarking agentic broad info-seeking. In *The Fourteenth International Conference on Learning Representations*.

Xixi Wu, Kuan Li, Yida Zhao, Liwen Zhang, Litu Ou, Huifeng Yin, Zhongwang Zhang, Xinmiao Yu, Dingchu Zhang, Yong Jiang, and 1 others. 2025. Resum: Unlocking long-horizon search intelligence via context summarization. *arXiv preprint arXiv:2509.13313*.

Qwen: An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 23 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Zhipu-AI. 2026. GLM-5.1: Towards long-horizon tasks.

Peilin Zhou, Bruce Leon, Xiang Ying, Can Zhang, Yifan Shao, Qichen Ye, Dading Chong, Zhiling Jin, Chenxuan Xie, Meng Cao, Yuxin Gu, Sixin Hong, Jing Ren, Jian Chen, Chao Liu, and Yining Hua. 2025. BrowseComp-ZH: Benchmarking web browsing ability of large language models in chinese. *Preprint*, arXiv:2504.19314.

A Calibration Error

We adopt Expected Calibration Error (ECE) to measure the alignment between model confidence and actual accuracy. Predicted probabilities are partitioned into five equally spaced bins: $[0, 0.2)$, $[0.2, 0.4)$, $[0.4, 0.6)$, $[0.6, 0.8)$, and $[0.8, 1.0]$. ECE is defined as:

$$\mathrm{ECE} = \sum_{i=1}^{K} \frac{n_i}{N} \left|\mathrm{acc}(i) - \mathrm{conf}(i)\right| \qquad (2)$$

where K is the number of bins, N is the total number of samples, n_i is the number of samples in the i-th bin, $\mathrm{acc}(i)$ is the empirical accuracy of

the i-th bin, and $\text{conf}(i)$ is the average predicted confidence in the i-th bin. A lower ECE indicates better calibration between model confidence and actual performance.

B System Prompt

We adopt the same system prompt as BrowseComp to instruct all models. The prompt content is as follows:

```
Your response should be in the following
format:
Explanation: {your explanation for your
final answer}
Exact Answer: {your succinct, final answer}
Confidence: {your confidence score between
0% and 100% for your answer}
```

C Tool Definitions

All models are equipped with two tools: `search` and `browse`. Their definitions are as follows:

```
{"type": "function",
 "function": {
   "name": "search",
   "description": "Web search, using traditional
     search engines like google, complex
     questions need to be broken down into
     simple queries",
   "parameters": {
     "type": "object",
     "properties": {
       "query": {
         "type": "array",
         "items": {"type": "string"},
         "description": "List of search queries.
           You can search up to 5 queries at
           the same time."}
       },
     "required": ["query"]
   }
 }
}
```

```
{"type": "function",
 "function": {
   "name": "browse",
   "description": "Retrieve detailed content
     from one or more specified web pages by
     providing their URLs.",
   "parameters": {
     "type": "object",
     "required": ["url"],
     "properties": {
       "url": {
         "type": "array",
         "description": "List of urls. You can
           browse up to 3 webpages at the
           same time.",
         "items": {"type": "string"}
       }
     }
   }
 }
}
```

D Case Studies

We present one tree-structured and one graph-structured example from LoHoSearch.

Case 1 (Tree-structured): Consider the following facts. They all pertain to one album. Which album is it?
1. It includes singles from another album (Album A), which was created by a group formed on a reality competition television series that originally aired from the early 2000s to the mid-2010s; additionally, Album A includes a solo song by a person (Person D) who has one child, works as a pop singer and television host, and was born in the late 1970s.
2. It includes a track written by a person (Person B) who shares the same name as a person (Person E) who earned an award in the early 2000s; this award has been presented annually since the year 2000 and has an alternative name that includes an acronym for a community service initiative.
3. It includes tracks produced by a person (Person C) who studied an academic discipline historically defined by roughly three main methods—individual casework, social group work, and community intervention work—and from the late 20th century, many different methods derived from these three classic methods.
Answer: The Best of No Angels

Case 2 (Graph-structured): Who is described by the following conditions?
1. This person toured with a musical group (Group C). Group C performs in a music genre (Genre D) that is featured at a music festival (Festival E), where Group C was a performer in the early 2010s. The group also released an album (Album H) in the early 2010s, which includes a musical composition (Composition G) by them. Additionally, Group C performs at a recurring event (Event F), and this person has performed at Event F as well.
2. They recorded a comedy album at a recurring event (Event I). Event I featured Group C as a guest of honor in the late 2010s.
3. Furthermore, this person appeared as a guest on a podcast (Podcast B) and wrote and performed shows at a film festival (Festival J).
Answer: Joseph Scrimshaw